## A Method to Combine Linguistic Ontology-Mapping Techniques

van Hage, W.R.; Katrenko, S.; Schreiber, A.T.

***published in***
The Semantic Web -- ISWC 2005: 4th International Semantic Web Conference, Galway, Ireland, November 6-10, 2005. Proceedings
2005

***document version***
Peer reviewed version

# A Method to Combine Linguistic Ontology-Mapping Techniques

Willem Robert van Hage[1], Sophia Katrenko[2], and Guus Schreiber[3]

[1] TNO, Science and Industry,
wrvhage@few.vu.nl
[2] Free University Amsterdam, Computer Science
schreiber@cs.vu.nl
[3] University of Amsterdam, Informatics Institute
katrenko@science.uva.nl

**Abstract.** We discuss four linguistic ontology-mapping techniques and evaluate them on real-life ontologies in the domain of food. Furthermore we propose a method to combine ontology-mapping techniques with high Precision and Recall to reduce the necessary amount of manual labor and computation.

## 1 Introduction

Ontologies are widely used to provide access to the semantics of data. To provide integrated access to data annotated with different, yet related, ontologies, one has to relate these ontologies in some way. This is commonly done by cross-referencing concepts from these ontologies. In different contexts this practice is called ontology mapping, schema matching, or meaning negotiation. In the literature one can find surveys of the widely varying methods of automated ontology mapping. For instance, in the surveys done by Kalfoglou and Schorlemmer [5]; and Rahm and Bernstein [8]. The latter organized the methods hierarchically. The ontology-mapping methods we develop in this paper fall in the categories *schema-only based*, which means they work on the conceptual part of the ontology and not on the annotated individuals and *linguistic*, since we use the labels of the concepts. The techniques we use come from the field of *information retrieval* (IR).

The work in this paper is done within the scope of the Adaptive Information Disclosure (AID) project, which is part of the greater effort of the Dutch "Virtual Labs for e-Science" project (VL-e) [1]. The AID project focusses on facilitating access to domain-specific text corpora, in particular articles about food. When the semantics of data sources or the information needs are of increasing complexity old-fashioned information-retrieval systems can fail to deliver due to the following reasons:

[1] http://www.vl-e.nl

Y. Gil et al. (Eds.): ISWC 2005, LNCS 3729, pp. 732–744, 2005.

- Domain-specific terms can have homonyms in a different domain. For instance, "PGA" stands for "Polyglandular Autoimmune Syndrome" and the "Professional Golfers' Association".
- Synonyms used by different communities can be difficult to relate to each other. For instance, some refer to "stomach acid" with "Betaine HCl", others use "Hydrochloric Acid".
- Skewed term-frequency distributions can lead to failing weighting schemes. For instance, the term "cancer" occurs as frequently as some stop words in the medical MedLine corpus, but it is an important term.

Ontologies pave the way for new techniques to facilitate access to domain-specific data. Semantic annotation of text resources can help to subdue jargon. [6,10] Obviously accessing annotated data sources is not without problems of its own. In practice different data sources are often annotated with different ontologies.[2] In order to provide integrated access using multiple ontologies, some form of ontology mapping needs to be done.

Within AID we focus on food information corpora. This domain–like the medical domain–struggles with an information overload and jargon issues. For instance, everyday household terms are intermingled with names of proteins and other chemical compounds. This complicates the formulation of good search queries. In this paper we test the applicability of four automated ontology-mapping techniques on real-life ontologies in the domain of food and assess their practical use. Specifically we try to map the USDA Nutrient Database for Standard Reference, release 16 (SR-16) [3] onto the UN FAO AGROVOC thesaurus (AGROVOC) [4] using that yield RDFS [1] subClassOf relations. The four techniques we discuss are listed below.

1. Learn subclass relations between concepts from AGROVOC and SR-16 by querying Google for Hearst patterns. [4]
2. Learn subclass relations by extracting them from Google snippets returned by the same queries with the help of shallow parsing using the TreeTagger part-of-speech tagger. [9]
3. Learn subclass relations by extracting them from a semi-structured data source, the CooksRecipes.com Cooking Dictionary, with MINIPAR [7].
4. Use the Google hits method as a sanity check to filter the dictionary mining results.

In Section 2 we discuss some related work to give an impression of current practice in relation extraction. In Section 3 we describe the experimental set-up we used in which we tested the four mapping techniques. In Section 4 we describe the four techniques in great detail and discuss the acquired results. In Section 5 we propose a method for applying the techniques in practice and we show how much manual labor can be saved.

[2] We use the term ontologies to include light-weight ontologies such as vocabularies and thesauri.

[3] http://www.nal.usda.gov/fnic/foodcomp/Data/SR16/sr16.html

[4] http://www.fao.org/agrovoc

## 2 Related Work

Brin proposed a method called Dual Iterative Pattern Relation Extraction (DIPRE) in his paper from 1998 [2]. He tested the method on part of his Google corpus–which at the time consisted of about 24 million web pages–to learn patterns that link authors to titles of their books. These patterns were then used to retrieve author-title relation instances from the same corpus. An example of such a pattern is the HTML bit: "`<li><b>`*title*`</b>` by *author*".

In 1992 Hearst devised a set of lexico-syntactic patterns for domain aspecific hyponym extraction [4]. His patterns found entrance in many applications such as Cimiano and Staab's PANKOW system. [3] The first method we discuss in this paper is similar to their work.

In their 2004 paper Cimiano and Staab try to accomplish two things. The first is a instance classification task: to classify geographical entities such as Amsterdam (City), Atlantic (Ocean), etc. The second is a subclass learning task: to reconstruct a subclass hierarchy of travel destinations mentioned in the LonelyPlanet website[5]. The method they use is the same for both tasks. They send Hearst patterns describing the relation they want to test to the Google API and depending on the number of hits Google returns they accept of reject the relation. For instance, the query "cities such as Amsterdam" yields 992 hits. Depending on which threshold they put on the number of hits they achieved Precision between .20 and .35 and Recall somewhere between .15 and .08. The higher the threshold, the higher the Precision and the lower Recall.

What we want to accomplish is a bit more complicated than either of Cimiano and Staab's tasks for two reasons. The food domain is less well-defined than the geographical domain, in which there are exhaustive thesauri such as TGN. The relations between the concepts are clearly defined. Countries have exactly one capital. Countries can border each other, etc. In the food domain such consensus does not exist. This means the evidence for relations that can be found in Google can be expected to be more ambiguous in the food domain than in the geographical domain.

## 3 Experimental Set-Up

Our set-up consists of the two thesauri we want to connect, the auxiliary sources of knowledge we use to learn the mappings from, and a gold-standard mapping to assess the quality of the learnt relations. In Section 3.3 we discuss the gold standard and the evaluation measures we use.

### 3.1 Thesauri

*AGROVOC.* This is a multi-lingual thesaurus made by the Food and Agriculture Organization of the United Nations (FAO). It consists of roughly 17,000

[5] `http://lonelyplanet.com/destinations`

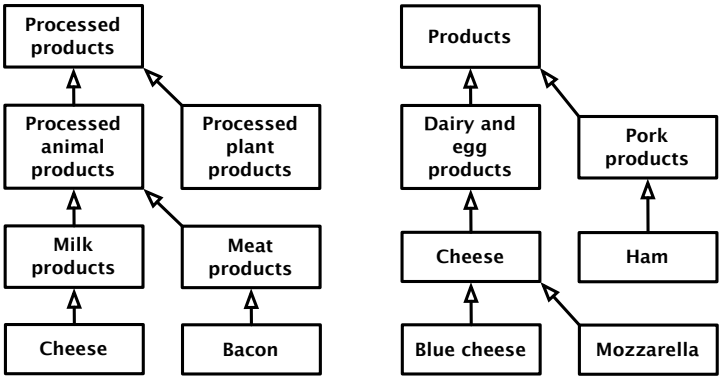


**Fig. 1.** excerpts from AGROVOC (left) and SR-16 (right)

concepts and three types of relations derived from the ISO thesaurus standard: use (preferred term), rt (related term) and bt (broader term). We use a RDFS version of this thesaurus where the broader term relation is represented with the RDFS subClassOf relation. The maximum depth of AGROVOC's subclass hierarchy is eight. Figure 1 shows an excerpt from AGROVOC. The text boxes are classes with their names and the arrows stand for subclass relations.

*SR-16.* This is the Nutrient Database for Standard Reference version 16 (SR-16) made by the United States Department of Agriculture (USDA), converted to RDFS and OWL by the AID group. It consists of roughly 6500 concepts and one relation, RDFS subClassOf. The maximum depth of the subclass hierarchy of SR-16 is four. Figure 1 shows an excerpt from SR-16.

### 3.2 Auxiliary Knowledge Sources

We used one general and one domain-specific source. The general source is Google and the domain-specific source is the CooksRecipes.com's Cooking Dictionary.

*Google.* Google [6] is an open domain search engine. At the moment (mid 2005) Google indexes more than 8 billion pages. The large size of Google allows makes it possible to use it for statistical comparison of words. Google has a programming interface called the Google API, that at the moment allows researchers to pose 1000 queries per day.

*CooksRecipes.com's Cooking Dictionary.* The CooksRecipes.com Cooking Dictionary provides definitions for ingredients, culinary terms and cooking techniques. It contains 1076 definitions. An example entry is: "**Basmati** an aged, aromatic long-grain rice grown in the Himalayan foothills; has a creamy yellow color, distinctive sweet, nutty aroma and delicate flavor..."

[6] `http://www.google.com`

### 3.3 Evaluation Method

In order to do full evaluation of the quality of a mapping between AGROVOC and SR-16 one would have to assess all possible subclass relations between a thesaurus of roughly 17,000 and one of around 6500 classes. This sums up to something of the order of hundreds of millions of possible mapping relations. With smart pruning of the possible mapping this still would have left us with more work than time allowed. Therefor we took samples from both thesauri on a common topic. From SR-16 we took one set of concepts about meats, containing the parts about beef, pork and poultry (chicken, turkey bacon, ham, etc.). From AGROVOC we took two sets of concepts, one containing the part about animal products (minced meat, cheese, leather, etc.), and one containing the part about food categories (processed foods, instant foods, snack foods, etc.).

For the experiments with Google we created a gold standard mapping by hand from the set of SR-16 concepts to both sets of AGROVOC concepts. The size of the mapping from meats to animal products is 31 relations out of 3696 possible relations. The size of the mapping from meats to food categories is 32 relations out of 792 possible relations.

The experiments with the CooksRecipes.com Dictionary yielded few results, distributed evenly over the thesauri, which made it hard to choose a subset of the thesaurus that contained a reasonable number of mapping relations. Therefor, we evaluated only the returned results. This means we are unable to say anything about Recall of the techniques using the CooksRecipes.com Dictionary.

The measures we used are Precision, Recall and F-Measure as used throughout the literature.[7] The F-Measure we use gives Precision and Recall an equal weight.

*Protocol.* The protocol we used can be summarized as follows: All concepts are to be interpreted in their original context. For instance, in AGROVOC chicken is a subclass of product, which means none of the individuals of the chicken class are live chickens. Taking this into account chicken is not a subclass of frozen foods, because some chicken products are never frozen, but chicken is a subclass of poultry, because all chicken products qualify as poulty.

## 4 Experiments

### 4.1 Hearst Patterns and Google Hits

The mapping technique described in this section is approximately the same as Cimiano and Staab's "Learning by Googling" method. It derives relations from Google hit counts on certain queries.

*Method*

1. **Create hypothetical relations between pairs of concepts from both thesauri.** For this experiment we chose to investigate all possible relations

[7] http://en.wikipedia.org/wiki/Information_Retrieval

**Table 1.** Hearst patterns used in this paper

| | | |
|---|---|---|
| | $concept_1$ such as | $concept_2$ |
| such | $concept_1$ as | $concept_2$ |
| | $concept_1$ including | $concept_2$ |
| | $concept_1$ especially | $concept_2$ |
| | $concept_1$ and other | $concept_2$ |
| | $concept_1$ or other | $concept_2$ |

from any of the concepts in the predefined set of SR-16 concepts to any of the concepts in both of the predefined sets of AGROVOC concepts (see Section 3.3).

2. **Construct Google queries containing Hearst patterns for each pair of concepts.** We chose to use the same Hearst patterns as Cimiano and Staab [3] except the apposition and copula patters, to reduce the number of Google queries, because these patterns did not yield enough results to be useful. The patterns are listed in the Table 1. Since we are only interested in the combined result of all the patterns we can further reduce the number of queries by putting the patterns in a disjunction. We chose the disjunction to be as long as possible given the limit Google imposes on the number of terms in a query (which was 32 at the time).
3. **Send the queries to the Google API.**
4. **Collect the hit counts for all Heart patterns that give evidence for the existence of a relation.** For instance, add the hits on the queries "milk products such as cheese", "milk products including cheese", etc. Since all these hits give a bit of evidence that cheese is a subclass of milk products.
5. **Accept all hypothetical relations that get more hits than a certain threshold value.** Reject all others.

*Results.* The average number of hits for the mapping to food categories is about 2.5 and to animal products it is about 1.3. Only about 2.5% of the patterns had one or more hits. The maximum number of hits we found was in the order of 1000, while Cimiano and Staab find hit counts in the order of 100,000. We suspect that this is the case because people rarely discuss the ontological aspects of food, because it is assumed to be common knowledge–everybody knows beef is a kind of meat–and hence can be left out. Since the total number of hits is so low we chose not to use a threshold, but to accept all relations that had one or more hits instead. Precision and Recall are shown in Table 2.

**Table 2.** Results of the Google hits experiment

| | Precision | Recall | F-Measure |
|---|---|---|---|
| to animal products | .17 (10/58) | .32 (10/31) | .22 |
| to food categories | .30 (17/56) | .53 (17/32) | .38 |

*Discussion.* The performance of the PANKOW system of Cimiano and Staab on geographical data is a Precision of .40 with a Recall of around .20 for instance classification and a Precision of .22 and a Recall of .16 for subclass extraction.

Overall Recall seems to be less of a problem in the food domain than in the geographical domain. The decent Recall values can be explained by the large size of the current Google corpus. On simple matters it is quite exhaustive. Even though the total hit counts in the food domain are lower than in the geographical domain it seems that a greater percentage of the relations is mentioned in Google. Apparently not all LonelyPlanet destinations have been discovered by the general web public. If you are interested in really high Recall in the field of geography you can simply look up your relations in the Getty Thesaurus of Geographic Names (TGN) [8].

Precision of the mapping to animal products seems to be comparable to the subclass learning task Cimiano and Staab set for themselves. The overall low Precision can be explained by the fact that when you use Google as a source of mappings between two thesauri you turn it from one into two mapping problems: from the thesaurus to Google; and then from Google to the other thesaurus. That means you have to bridge a vocabulary gap twice and hence introduce errors twice.

Precision of mapping to food categories using Google hits seems to be comparable to that of instance classification. Mapping to animal products, i.e. mapping between concepts of similar specificity, appears to be more difficult.

### 4.2 Hearst Patterns and Google Snippets

The second mapping technique is a modification of the previous technique. Instead of deriving relations from Google hit counts we analyze the snippets presented by Google that summarize the returned documents. We try to improve performance by shallow parsing the context of the occurrence of the Hearst pattern and remove false hits.

*Method*

1. **Follow step 1 through 3 from the "Hearst patterns and Google hits" method.**
2. **Collect all the snippets Google returns.** Snippets are the short exerpts from the web pages that show a bit of the context of the query terms.
3. **Extract the patterns.** To accomplish this we part-of-speech tag the snippets with TreeTagger and recognize sequences of adjectives and nouns as concept names. Then we try find all Hearst patterns over the concept names in the snippets.
4. **Discard all patterns that contain concept names that do not exactly match the original concept names.** For instance, if the original pattern looked like "soup such as chicken", discard the matches on "soup such as chicken soup", because these give false evidence for the relation

[8] `http://www.getty.edu/research/conducting_research/vocabularies/tgn`

chicken subClassOf soup. We ignore prefixes to the concept names from the following list: "other", "various", "varied", "quality", "high quality", "fine", "some", and "many". This unifies concept names such as "meat products" and "high quality meat products".

5. **Count every remaining occurrence of the pattern as evidence that the relation holds.**
6. **Follow step 4 and 5 from the "Hearst patterns and Google hits" method.**

*Results.* Analysis of the snippets improves Precision while sacrificing Recall. Overall performance indicated by the F-Measure does not chance much. Shallow parsing the snippets removed many false hits. For instance, "salads such as chicken salad" does not lead to chicken subClassOf salad anymore. The exact Precision and Recall are shown in Table 3.

**Table 3.** Results of the Google snippets experiment

| | Precision | Recall | F-Measure |
|---|---|---|---|
| to animal products | .38 (7/18) | .22 (7/31) | .27 |
| to food categories | .50 (12/24) | .37 (12/32) | .42 |

*Discussion.* Even the Precision achieved with mapping to concepts of similar specificity (to animal products) is comparable to the level PANKOW achieves for instance classification. The mapping to food categories, which is closer to the instance classification task, now achieves a higher Precision and Recall than PANKOW.

As Cimiano and Staab noted downloading the whole documents for analysis could further improve the results. This might even improve Recall a bit if these documents contain more good Hearst patterns than those that caused them to appear in Google's result set.

### 4.3 Extraction from a Dictionary

With the third mapping technique we try to exploit the implicit editor's guidelines of a dictionary to achieve an even higher grade of Precision than the Google Snippets technique described in the previous section. As an example we took a dictionary that includes terms from both thesauri, the CooksRecipes.com Cooking Dictionary. This dictionary is relatively small compared to the thesauri, but it covers about the same field as SR-16.

*Method*

**Find regularities in the dictionary that highly correlate with subclass relations.** We found that the editor of the dictionary often starts a definition with the superclass of the described concept. The following steps are tailored to exploit this regularity.

1. **Select all entries describing a concept that literally matches a concept from AGROVOC or SR-16.**
2. **Parse the entry with MINIPAR.**
3. **Extract the first head from the parse tree.** For instance, the entry of the concept basmati starts with "an aged, aromatic long-grain rice grown in ..." The first head in this sentence is "rice".
4. **Check if the first head corresponds to a concept in the other thesaurus** If basmati is a concept from AGROVOC, try to find the concept rice in SR-16 and vice versa.
5. **Construct a subclass relation between the concept matching the entry name and the one matching the first head.**

*Results.* More than half of all the returned relations, even those failing the check in step 4, are correct subclass relations according to our strict evaluation protocol. As expected, given the relatively wide scope of the dictionary, step 4 eliminates most of the results. However the mapping relations that are left are of high quality. The exact results are shown in Table 4.

**Table 4.** Results of the dictionary extraction experiment

| | Precision |
|---|---|
| relations not forming a mapping | .53 (477/905) |
| mapping entire AGROVOC–SR-16 | .75 (16/21) |

*Discussion.* We exploited a regularity in the syntax of the data. This yields high Precision results. Clearly, Recall of this method is dependent on the size of the dictionary and the overlap between the dictionary and the thesauri.

We noticed that most of the errors could have been filtered out by looking for evidence on Google. For instance, the entry: "**leek** a member of the lily family *(Allium porrum)*; ..." would cause our technique to suggest the relation leek subClassOf member. One query could have removed this false relation from the result list, because "member such as leek" gives no hits on Google.

### 4.4 Combination of Google Hits and Dictionary Extraction

The fourth technique is an improvement to the dictionary extraction technique. We use the Google hits technique to filter false relations out of the list of results provided by extraction.

*Method*

1. **Follow all the steps of the Dictionary Extraction method.** This yields a list of relations.
2. **For each extracted relation follow step 2–5 from the Google hits method.** This filters out all relations for which no evidence can be found on Google using Hearst patterns.

*Results.* Applying the Google hits technique as a sanity check on the extraction results greatly reduces the number of relations. Precision of this smaller result set is higher than with both the Google hits and dictionary extraction technique. Around 63% of the correct results were removed versus 92% of the incorrect results. The results are shown in Table 5.

**Table 5.** Results of combining dictionary extraction and Google hits

| | Precision |
|---|---|
| relations not forming a mapping | .53 (477/905) |
| after Google hits sanity check | .84 (178/210) |
| mapping entire AGROVOC to SR-16 | .75 (16/21) |
| after Google hits sanity check | .94 (15/16) |

*Discussion.* The combination of Google hits and a dictionary gave the best Precision of the four techniques. Most of the mismatches caused by definitions that did not exactly fit the regularity that we exploited with the dictionary extraction technique were removed by applying the Google hits technique. On the other hand, a substantial portion of the correct results was also removed.

We noticed that most of the incorrect relations that were not removed are easily recognizable by hand. If the superclass is not directly food related the relation is usually false. For instance, mayonnaise subClassOf cold. Most relations to latin names of plants were inverted. For instance, rosmarinus officinalis subClassOf rosemary. There is another member of the rosemary family, "Rosmarinus eriocalix", so rosmarinus officinalis should be a subclass.

## 5 Method Proposal

As we discussed in Section 3.3 simply checking all possible relations between two ontologies is task of quadratic complexity. In theoretical computer science this might qualify as a polynomial with a low degree, but for a mapping technique that uses the Google API (which only allows 1000 queries per account per day) this means it does not scale well. Furthermore, assessing a quadratic number of relations by hand is often not feasible. Therefor we propose to combine high Precision techniques and techniques that achieve a high Recall per human assessment. The method we propose is as follows:

1. **Find a small set of high Precision mapping relation as starting points, preferably distributed evenly over the ontologies.** This could be done with the last two techniques we described or with tools such as PROMPT [9]. Which technique works best depends largely on the naming conventions used in the ontologies.
2. **Manually remove all the incorrect relations.** Assessing the results of the dictionary extraction technique took about one man hour.

[9] `http://protege.stanford.edu/plugins/prompt/prompt.html`

3. **For each correct relation select the concepts surrounding the subject and object concepts.** For instance, if the SR-16 concept cheese (see Figure 1) was correctly mapped as a subclass of the AGROVOC concept Milk products, one would select a subclass tree from SR-16 that contains cheese and a subclass tree from AGROVOC that contains Milk products. This can be accomplished in the following two steps:
   (a) **Travel up the subclass hierarchy from the starting point.** Go as far as possible as long as it is still clear what is subsumed by the examined concept, without having to examine the subtrees of the sibling concepts. A suitable top concept from SR-16 could be Dairy and egg products because it is immediate clear to us what is subsumed by this concept without having to look at the Pork products concepts. A suitable top concept from AGROVOC could be Processed animal products.
   (b) **Select all subclasses of the two top concepts.** Collect the concepts as two sets.

   This could be done using tools such as Triple20 [10] or Sesame [11].
4. **Find relations between the two sets of concepts returned in the previous step.** This could be done with the Google snippets technique.
5. **Manually remove all incorrect relations.** The evaluation of the mapping between the AGROVOC animal product concepts and the SR-16 meat concepts took us four man hours. Assessing all the mappings returned by the previous steps could take days. The higher the applied mapping techniques' Precision, the less time this step takes.
6. **Manually add all omissions.** Creating a list of omissions during the assessments of the previous step reduces the amount of work in this step. The higher the applied mapping techniques' Recall, the less time this step takes.

This method reduces the search space by eliminating cross-references between concepts in unrelated parts of the ontologies. For instance, possible relations between concepts in the part of AGROVOC about legumes and in the part of SR-16 about poultry would be ignored if step 1 did not yield any relations between those parts. Hence the number of queries we have to send to Google is reduced along with the number of necessary manual assessments low.

## 6 Discussion

We discussed four ontology mapping techniques and evaluated their performance. There is a clear trade-off between Precision and Recall. The more assumptions we make the higher Precision gets and the lower Recall. We showed that exploiting syntactic information by using a part-of-speech tagger can improve Precision of ontology-mapping methods based on Google hits such as our Google hits method and possibly PANKOW.

[10] `http://www.swi-prolog.org/packages/Triple20`
[11] `http://www.openrdf.org`

We showed that in our experiments finding subclass relations to generic concepts such as food categories is easier than mapping concepts that are roughly equal in specificity. We hypothesize that this is because the former discriminate more clearly between different interpretations of concepts and are therefor used more often. For instance, the phrase "chickens such as roosters" is less discriminating about the meaning of the word "rooster" than "poultry such as roosters" or "birds such as roosters".

Furthermore, we introduced a method that extends the PANKOW two-step method by Cimiano and Staab to decrease the number of necessary Google queries and the amount of manual work.

## Acknowledgements

This paper has benefitted from input from the AID group's participants: Pieter Adriaans, Jan van Eijck, Leonie IJzereef, Machiel Jansen, Hap Kolb, Maarten de Rijke and the authors of this paper. Sophia Katrenko provided the RDFS and OWL version of SR-16. We want to thank Marco Roos and Scott Marshall from the Micro Array Department of the University of Amsterdam, Michel Klein at the Computer Science department of the Free University Amsterdam for valuable discussions, Victor de Boer who organized the Ontology Learning and Population Workshop at the Human-Computer Studies Laboratory of the University of Amsterdam and and everybody who attended. Furthermore we want to thank Thijs de Graaf, Wessel Kraaij and Dolf Trieschnigg at the Signal Processing group at TNO Science and Industry.

## References

1. Dan Brickley and Ramanathan Guha. *Resource description framework (RDF) schema specification 1.0.* W3C, March 2000.
2. Sergey Brin. Extracting patterns and relations from the world wide web. In *WebDB Workshop at 6th International Conference on Extending Database Technology, EDBT'98*, 1998.
3. Philipp Cimiano and Steffen Staab. Learning by googling. *SIGKDD Explor. Newsl.*, 6(2):24–33, 2004.
4. Marti Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th International Conference on Computational Linguistics*, 1992.
5. Yannis Kalfoglou and Marco Schorlemmer. Ontology mapping: the state of the art. *The knowledge engineering review*, 18(1):1–31, march 2003.
6. Jaap Kamps. Improving retrieval effectiveness by reranking documents based on controlled vocabulary. In *Advances in Information Retrieval: 26th European Conference on IR Research (ECIR)*, 2004.
7. Dekang Lin. Dependency-based evaluation of minipar. In *Proceedings of the Workshop on the Evaluation of Parsing Systems, First International Conference on Language Resources and Evaluation*, Granada, Spain, May 1998.
8. Erhard Rahm and Philip A. Bernstein. A survey of approaches to automatic schema matching. *VLDB Journal*, 10(4), 2001.

9. Helmut Schmid. Probabilistic part-of-speech tagging using decision trees. In *Proc. of International Conference on New Methods in Language Processing*, 1994.
10. H. Stuckenschmidt, F. van Harmelen, A. de Waard, T. Scerri, R. Bhogal, J. van Buel, I. Crowlesmith, Ch. Fluit, A. Kampman, J. Broekstra, and E. van Mulligen. Exploring large document repositories with rdf technology: The dope project. *IEEE Intelligent Systems*, 19(3):34–40, 2004.